Computers & Security, 8 (1989) 35–41

Refereed Article

A Mathematical Theory for the Spread of Computer Viruses

Dr. Winfried Gleissner

Industrieanlagen-Betriebsgesellschaft mbH, Ottobrunn, F.R.G.

A model is introduced to treat the spread of computer viruses mathematically. A recurrence formula is given which allows a closed expression to be derived for the probability that, starting from an initial state, a given viral state will be reached after executing exactly k programs. In some special cases this recurrence formula can be used for numeric computations. It is shown that the infection process does not stop before all programs are infected, which are visible for any infected program in the initial state.

Keywords: Computer viruses, Mathematical model of virus infection

1. The Situation

A computer virus is a program that can reproduce itself and modify other programs by including a possibly evolved copy of itself [1]. That means that whenever an infected program is called, the virus is implanted into another program, if there is any, of the account from where it was called. In refs. 1 and 2 some experiments with this kind of program are described. The result was that within a few hours the whole computer system was infected. The aim of this paper is to develop a closed expression for the probability that any given program is infected after a given time. To achieve this aim it must be known which programs were initially infected and for each program one needs the probability that it is called. The use of a computer is regarded as a sequence of program calls. For the purpose of the present paper there is no difference between the call of a system command, a standard program (editor, linkage editor, compiler), and a user-written program.

2. The Notation

On the computer system there are N accounts. The m_i programs in account i are denoted by $P_1^i, P_2^i, \ldots, P_{m_i}^i$. Let $q_{ij}^{(k)}$ denote the probability that user i calls the program P_j^k in the account k. This gives

$$\sum_{i=1}^{N} \sum_{k=1}^{N} \sum_{j=1}^{m_k} q_{ij}^{(k)} = 1$$

Let F denote the set of all N-vectors with integer components, which is defined as follows:

$$F = \{\nu \in N_0^N;\ 0 \leqslant \nu_i \leqslant m_i,\ 1 \leqslant i \leqslant N\}$$

For $\nu \in F$ the computer system is said to be in viral state ν, if in account i there are ν_i infected programs. One can further assume that these are the first ν_i programs according to the alphabetic order in which the programs are written down. For convenience the viral state in which all programs are infected is denoted by $\nu^{(F)}$, i.e.

$$\nu^{(F)} = (m_1, \ldots, m_N)^T$$

On F one defines a modulus by

$$|v| = \sum_{i=1}^{N} v_i$$

and an order by

$$v \geqslant \bar{v} \Leftrightarrow v_i \geqslant \bar{v}_i \qquad 1 \leqslant i \leqslant N$$

Given the viral degree $v \in F$, one needs the probability $s_{i,v}$ that an infected program will be called from account i

$$s_{i,v} = \sum_{k=1}^{N} \sum_{j=1}^{v_k} q_{ij}^{(k)}$$

The probability t_v that any not infected program is executed is calculated as

$$t_v = \sum_{i=1}^{N} \sum_{k=1}^{N} \sum_{j=v_k+1}^{m_k} q_{ij}^{(k)}$$

For purposes which will become clear later, one defines for $v_i = m_i$ and for $v = v^{(F)}$

$$s_{1,v} = 0 \qquad t_{v^{(F)}} = 1$$

Let $p_{\bar{v},k}^{(v)}$ denote the probability that starting from viral state v viral state $\bar{v}$ will be reached after exactly k program calls

3. The Recursion Formulae

The $p_{\bar{v},k}^{(v)}$ can be calculated recursively as follows:

$$p_{\bar{v},0}^{(v)} = \begin{cases} 1 & v = \bar{v} \\ 0 & \text{otherwise} \end{cases}$$

For $\bar{v} = (\bar{v}_1, \bar{v}_2, \ldots, \bar{v}_N)^T$ $\bar{v}^{(i)}$ denotes the vector whose ith component is reduced by unity

$$\bar{v}^{(i)} = (\bar{v}_1, \ldots, \bar{v}_{i-1}, \bar{v}_i - 1, \bar{v}_{i+1}, \ldots, \bar{v}_N)^T$$

Using this notation one obtains

$$p_{\bar{v},k}^{(v)} = t_{\bar{v}} \text{Prob}\{\bar{v} \text{ is reached after } k\text{-1 calls already}\}$$

$$+ \text{Prob}\{\text{the last program is infected by the } k\text{th call}\}$$

$$p_{v,k}^{(v)} = t_{\bar{v}} p_{\bar{v},k-1}^{(v)} + \sum_{i=1}^{N} s_{i,\bar{v}^{(i)}} p_{\bar{v}^{(i)},k-1}^{(v)} \qquad (1)$$

The following recursion formula holds for the $p_{\bar{v},k}^{(v)}$.

Lemma 1:

$$p_{\bar{v},k}^{(v)} = \sum_{i=1}^{N} s_{i,\bar{v}^{(i)}} \sum_{j=0}^{k-1} t_{\bar{v}}^{\,j} p_{\bar{v}^{(i)},k-1-j}^{(v)}$$

Proof: The assertion is proven by induction on k. For $k = 1$

$$p_{\bar{v},1}^{(v)} = t_{\bar{v}} p_{\bar{v},0}^{(v)} + \sum_{i=1}^{N} s_{i,\bar{v}^{(i)}} p_{\bar{v}^{(i)},0}^{(v)}$$

and as $p_{\bar{v},0}^{(v)} = 0$

$$p_{\bar{v},1}^{(v)} = \sum_{i=1}^{N} s_{i,\bar{v}^{(i)}} \sum_{j=0}^{0} t_{\bar{v}}^{\,j} p_{\bar{v}^{(i)},0-j}^{(v)}$$

Induction from k to $k+1$

$$p_{\bar{v},k+1}^{(v)} = t_{\bar{v}} p_{\bar{v},k}^{(v)} + \sum_{i=1}^{N} s_{i,\bar{v}^{(i)}} p_{\bar{v}^{(i)},k}^{(v)}$$

$$= t_{\bar{v}} \left(\sum_{i=1}^{N} s_{i,\bar{v}^{(i)}} \sum_{j=0}^{k-1} t_{\bar{v}}^{\,j} p_{\bar{v}^{(i)},k-1-j}^{(v)} \right) + \sum_{i=1}^{N} s_{i,\bar{v}^{(i)}} p_{\bar{v}^{(i)},k}^{(v)}$$

$$= \sum_{i=1}^{N} s_{i,\bar{v}^{(i)}} \sum_{j=0}^{k} t_{\bar{v}}^{\,j} p_{\bar{v}^{(i)},k-j}^{(v)}$$

This ends the proof of the lemma.

4. The Calculation of the Infection Probabilities

For $\bar{v} \geqslant v$ one defines

$$l = |\bar{v}| - |v| = \sum_{i=1}^{N} (\bar{v}_i - v_i)$$

Let $F_{v,\bar{v}}$ denote the set of all sequences of length l of vectors $w = (\vec{w}^{(i)})_{0 \leqslant i \leqslant l}$ with

$$v = \vec{w}^{(0)} \leqslant \vec{w}^{(1)} \leqslant \ldots \leqslant \vec{w}^{(i)} \leqslant \vec{w}^{(i+1)} \leqslant \ldots$$

$$\leqslant \vec{w}^{(l-1)} \leqslant \vec{w}^{(l)} = \bar{v}$$

with

$$|\vec{w}^{(i)}|+1=|\vec{w}^{(i+1)}| \qquad 0\leqslant i\leqslant l-1$$

If $\vec{w}^{(\lambda)}$ and $\vec{w}^{(\lambda+1)}$ differ in the i_λth component then

$$s_{i_\lambda \vec{w}^{(\lambda)}}=\sum_{k=1}^{N}\sum_{j=1}^{\vec{w}_k^{(\lambda)}} q_{i_\lambda j}{}^{(k)}$$

For an element $w\in F_{v,\vec{v}}$ one defines the Vandermond determinant V_w by

$$V_w=\begin{vmatrix} 1 & 1 & \cdots & 1 & 1 \\ t_{\vec{w}^{(0)}} & t_{\vec{w}^{(1)}} & & t_{\vec{w}^{(l-1)}} & t_{\vec{w}^{(l)}} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ t_{\vec{w}^{(0)}}{}^{l-1} & t_{\vec{w}^{(1)}}{}^{l-1} & \cdots & t_{\vec{w}^{(l-1)}}{}^{l-1} & t_{\vec{w}^{(l)}}{}^{l-1} \\ t_{\vec{w}^{(0)}}{}^{l} & t_{\vec{w}^{(1)}}{}^{l} & \cdots & t_{\vec{w}^{(l-1)}}{}^{l} & t_{\vec{w}^{(l)}}{}^{l} \end{vmatrix}$$

(For more details see ref [3] p. 179.)

If the sequence w' is obtained from the sequence w by deleting the last element, the following recursion formula holds:

$$V_w=V_{w'}\prod_{\lambda=0}^{l-1}\left(t_{\vec{w}^{(l)}}-t_{\vec{w}^{(\lambda)}}\right) \qquad (2)$$

For $0\leqslant j\leqslant 1$ one obtains the determinant $V_{w,(j)}$ deleting the last line and the jth column of V_w. The following determinant $V_{w,k}$ is derived from the Vandermond form substituting the exponent k for the exponent l in the last line:

$$V_{w,k}=\begin{vmatrix} 1 & 1 & \cdots & 1 & 1 \\ t_{\vec{w}^{(0)}} & t_{\vec{w}^{(1)}} & & t_{\vec{w}^{(l-1)}} & t_{\vec{w}^{(l)}} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ t_{\vec{w}^{(0)}}{}^{l-1} & t_{\vec{w}^{(1)}}{}^{l-1} & \cdots & t_{\vec{w}^{(l-1)}}{}^{l-1} & t_{\vec{w}^{(l)}}{}^{l-1} \\ t_{\vec{w}^{(0)}}{}^{k} & t_{\vec{w}^{(1)}}{}^{k} & \cdots & t_{\vec{w}^{(l-1)}}{}^{k} & t_{\vec{w}^{(l)}}{}^{k} \end{vmatrix}$$

and if $l=0$

$$V_{w,k}=t_{\vec{w}^{(0)}}{}^{k}$$

The $V_{w,k}$ can be calculated using the $V_{w(j)}$ by the following lemma:

Lemma 2:

$$V_{w,k}=\sum_{j=0}^{l-1}(-1)^{l+1-j}\left(t_{\vec{w}^{(l)}}{}^{k}-t_{\vec{w}^{(j)}}{}^{k}\prod_{\substack{i=0\\ i\neq j}}^{l-1}\left(t_{\vec{w}^{(l)}}-t_{\vec{w}^{(i)}}\right)V_{w'(j)}\right.$$

Proof: Adding the first line times $t_{\vec{w}^{(l)}}{}^{k}$ to the negative of the last line one obtains

$$V_{w,k}=(-1)\begin{vmatrix} 1 & 1 & \cdots & 1 & 1 \\ t_{\vec{w}^{(0)}} & t_{\vec{w}^{(1)}} & & t_{\vec{w}^{(l-1)}} & t_{\vec{w}^{(l)}} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ t_{\vec{w}^{(0)}}{}^{l-1} & t_{\vec{w}^{(1)}}{}^{l-1} & \cdots & t_{\vec{w}^{(l-1)}}{}^{l-1} & t_{\vec{w}^{(l)}}{}^{l-1} \\ t_{\vec{w}^{(l)}}{}^{k}-t_{\vec{w}^{(0)}}{}^{k} & t_{\vec{w}^{(l)}}{}^{k}-t_{\vec{w}^{(1)}}{}^{k} & \cdots & t_{\vec{w}^{(l)}}{}^{k}-t_{\vec{w}^{(l-1)}}{}^{k} & 0 \end{vmatrix}$$

$$=(-1)\sum_{j=1}^{l}(-1)^{2(l+1)-j}\left(t_{\vec{w}^{(l)}}{}^{k}-t_{\vec{w}^{(l-j)}}{}^{k}\right)V_{w(l-j)}$$

$$=\sum_{j=0}^{l-1}(-1)^{l+1-j}\left(t_{\vec{w}^{(l)}}{}^{k}-t_{\vec{w}^{(j)}}{}^{k}\right)V_{w(j)}$$

$$=\sum_{j=0}^{l-1}(-1)^{l+1-j}\left(t_{\vec{w}^{(l)}}{}^{k}-t_{\vec{w}^{(j)}}{}^{k}\prod_{\substack{i=0\\ i\neq j}}^{l-1}\left(t_{\vec{w}^{(l)}}-t_{\vec{w}^{(i)}}\right)V_{w'(j)}\right.$$

Now the following theorem can be proven without undue effort:

Theorem:

$$p_{\bar{v}\,k}^{(v)} = \sum_{w \in F_{v,\bar{v}}^{(l)}} \prod_{\lambda=0}^{l-1} s_{i_\lambda \vec{w}^{(\lambda)}} \frac{V_{w,k}}{V_w}$$

Proof: The assertion is proven by induction on the distance l of v and $\bar{v}$. For $l=0$

$$F_{v,v} = \{(v)\} \qquad V_{(v)} = 1 \qquad V_{(v),k} = t_v^k$$

yields

$$p_{v\,k}^{(v)} = t_v^k$$

Induction from $l-1$ to l By Lemma 1

$$p_{\vec{v}\,k}^{(v)} = \sum_{i=1}^{N} s_{i\,\vec{v}^{(l)}} \sum_{j=0}^{k-1} t_{\vec{v}}^{j} p_{\vec{v}^{(l)}\,k-1-j}^{(v)}$$

$$= \sum_{i=1}^{N} s_{i\,\vec{v}^{(l)}} \sum_{j=0}^{k-1} t_{\vec{v}}^{j} \sum_{w \in F_{v,\vec{v}^{(l)}}^{(l-1)}} \prod_{\lambda=0}^{l-2} s_{i_\lambda \vec{w}^{(\lambda)}} \frac{V_{w,k-1}}{V_w}$$

$$= \sum_{i=1}^{N} \sum_{w \in F_{v,\vec{v}}^{(l)}} \frac{s_{i,\vec{v}^{(l)}}}{V_w} \prod_{\lambda=0}^{l-2} s_{i_\lambda \vec{w}^{(\lambda)}} \sum_{j=0}^{k-1} t_{\vec{v}}^{j} V_{w\,k-1}$$

With Lemma 2 one infers

$$\sum_{j=0}^{k-1} t_{\vec{v}}^{j} V_{w\,k-1} = \frac{V_{(w,\vec{v}),k}}{\prod_{j=0}^{l-1} (t_{\vec{w}^{(l)}} - t_{\vec{w}^{(j)}})}$$

and with this and eqn (2)

$$p_{\vec{v}\,k}^{(v)} = \sum_{i=1}^{N} s_{i\,\vec{v}^{(l)}} \sum_{w \in F_{v,\vec{v}}^{(l)}} \prod_{\lambda=0}^{l-2} s_{i_\lambda \vec{w}^{(\lambda)}} \frac{V_{(w,v),k-1}}{V_{(w,v)}}$$

$$= \sum_{w \in F_{v,\vec{v}}^{(l)}} \prod_{\lambda=0}^{l-1} s_{i_\lambda \vec{w}^{(\lambda)}} \frac{V_{w,k}}{V_w}$$

This ends the proof of the theorem.

5. Conclusions

The conclusions will be stated in the form of three remarks.

Remark 1: Accounts that do not call any infected programs cannot be infected If for $v \in F$ there exists an i such that $s_{i\,v} = 0$ then for $\vec{v} \geqslant v$ with $v_i > v_i$

$$p_{\vec{v}\,k}^{(v)} = 0 \quad \text{for } \textit{all} \quad k \in N$$

Proof: As $\vec{v}_i > v_i$ there must be an infection in the account of the ith user, i.e. for every $w \in F_{v,\vec{v}}$ there must be an index λ such that $i_\lambda = i$. As

$$s_{i_\lambda \vec{w}^{(\lambda)}} = 0$$

$$p_{\vec{v}\,k}^{(v)} = \sum_{w \in F_{v,\vec{v}}^{(l)}} \prod_{\lambda=0}^{l-1} s_{i_\lambda \vec{w}^{(\lambda)}} \frac{V_{w,k}}{V_w} = 0$$

For $v \in F$ one defines

$$N_v = \{i;\ s_{i\,\vec{v}} = 0,\ \vec{v} \geqslant v\}$$

$$F_v = \{\vec{v} \in F;\ \vec{v} \geqslant v \quad \text{and} \quad \vec{v}_i = v_i \quad \text{for} \quad i \notin N_v\}$$

$$\vec{v}^{(v)} \in F_v \quad \textit{with} \quad v_i^{(v)} = m_i \quad \text{for} \quad i \notin N_v$$

Remark 2: If v is the initial viral state, the infection process does not stop before the viral state $\vec{v}^{v}$ is reached. One has to show that for $\vec{v} \in F_v$ with $\vec{v} \neq \vec{v}^{v}$

$$\lim_{k \to \infty} p_{\vec{v}\,k}^{(v)} = 0$$

Proof: As $s_{i\,v} > 0$, $t_{\vec{w}^{(l)}} < 1$ for all $w \in F_{v\,\vec{v}}$. If all $t_{\vec{w}^{(l)}}$ are distinct

$$\lim_{k \to \infty} V_{w\,k} = 0$$

If some $t_{\vec{w}^{(l)}}$ are equal, one can choose sequences $t_{\vec{w}_\varepsilon^{(l)}}$ with

$$\lim_{\varepsilon \to 0} t_{\vec{w}_\varepsilon^{(l)}} = t_{\vec{w}}^{(h)}$$

$$t_{\vec{w}^{(k)}} \neq t_{\vec{w}^{(k)}} \quad \text{for} \quad h \neq k$$

and

$$t_{\vec{w}^{(l)}} < 1$$

Using de l'Hopital's rule one calculates

$$\lim_{k \to \infty} \frac{V_{w,k}}{V_w} = \lim_{k \to \infty} \lim_{\varepsilon \to 0} \frac{V_{w_\varepsilon k}}{V_{w_\varepsilon}} = \lim_{\varepsilon \to 0} \lim_{k \to \infty} \frac{V_{w_\varepsilon k}}{V_{w_\varepsilon}} = 0$$

and hence

$$\lim_{k \to \infty} p_{\vec{v}\,k}^{(v)} = 0$$

Remark 3: The maximum possible viral degree $\bar{v}^{(F)}$ is actually reached in the limit

If $\bar{v} = \bar{v}^{(F)}$

$$\lim_{k \to \infty} p_{\bar{v},k}^{(v)} = 1$$

Proof: The case where some $t_{\bar{w}^{(h)}}$ are equal can be treated as the limit of the case where all $t_{\bar{w}^{(h)}}$ are distinct. Hence only this case will be considered. The assertion shall be proven by induction on the distance l between v and $\bar{v}$. For $l = 1$ $t_{\bar{w}^{(0)}} = 1$ and

$$V_w = 1 - t_{\bar{w}^{(0)}} = s_{i_0\,\bar{w}^{(0)}}$$

Hence

$$\lim_{k \to \infty} p_{\bar{v},k}^{(v)} = 1$$

For the induction $v^{(i)}$ denotes the viral degree which is obtained from v by adding 1 to the ith component. The distance between $v^{(i)}$ and $\bar{v}^{(F)}$ is assumed to be l. Then

$$\lim_{k \to \infty} p_{\bar{v},k}^{(v)} = \lim_{k \to \infty} \sum_{w \in F_v\,\bar{v}} \prod_{\lambda=0}^{l} s_{i_\lambda\,\bar{w}^{(\lambda)}} \frac{V_{w,k}}{V_w}$$

$$= \lim_{k \to \infty} \sum_{i=1}^{N} s_{i\,v} \sum_{w \in F_v^{(i)}\,\bar{v}} \prod_{\lambda=0}^{l-1} s_{i_\lambda\,\bar{w}^{(\lambda)}} \frac{V_{(v,w),k}}{V_{(v\,w)}}$$

$$= \sum_{i=1}^{N} \frac{s_{i,v}}{1 - t_v} \sum_{w \in F_v^{(i)}\,\bar{v}} \frac{\prod_{\lambda=0}^{l-1} s_{i_\lambda\,\bar{w}^{(\lambda)}}}{\prod_{\lambda=0}^{l-1} 1 - t_{\bar{w}^{(\lambda)}}}$$

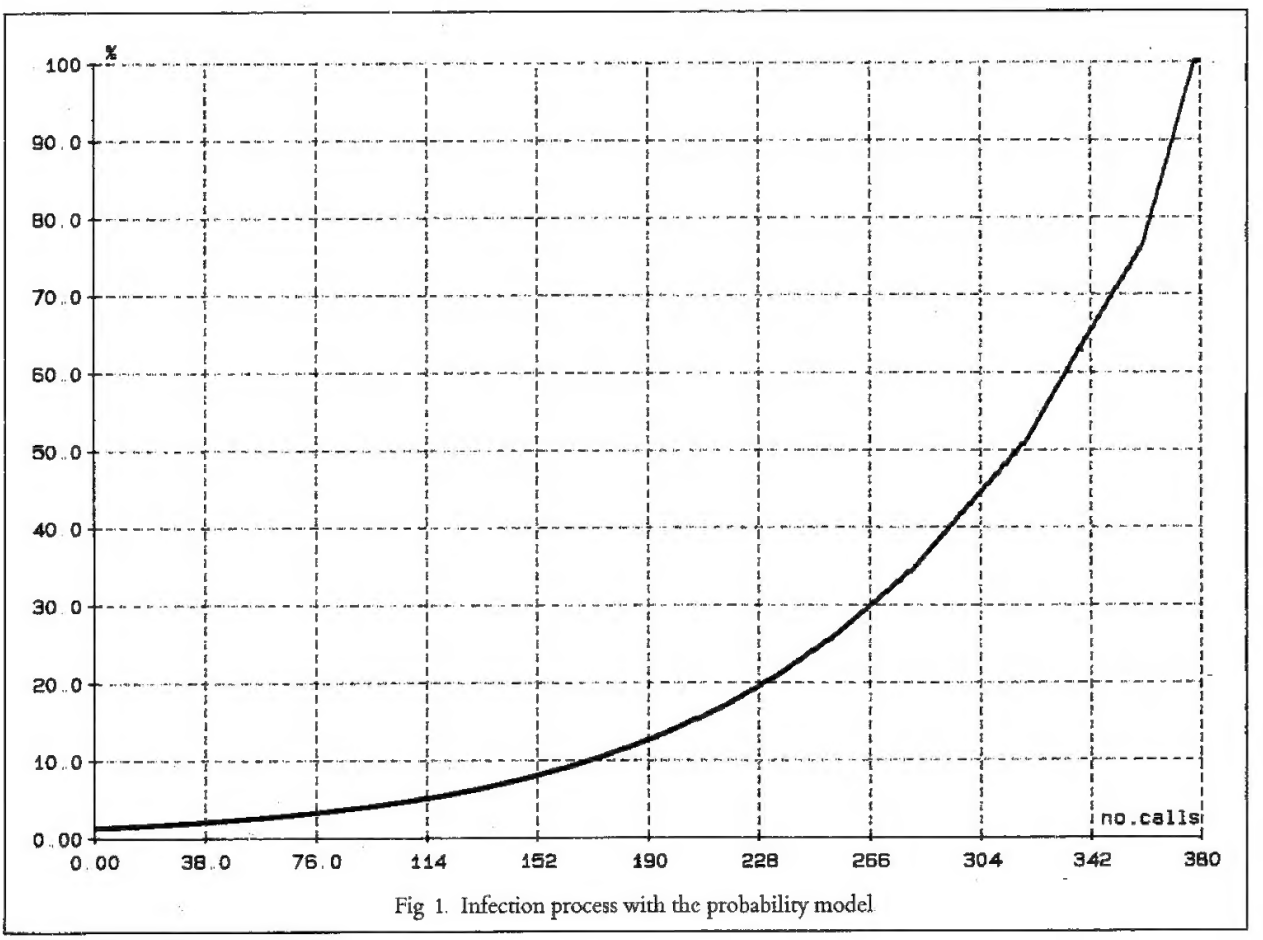


Fig 1. Infection process with the probability model

$$= \sum_{i=1}^{N} \frac{s_{i,v}}{1-t_v} \lim_{k\to\infty} p_{\bar{v},k}^{(v^{(l)})} = \sum_{i=1}^{N} \frac{s_{i,v}}{1-t_v} = 1$$

6. Numerical Examples

Number of infection paths: Using the notation introduced above one can calculate the number $I_N^{(v)}$ of possible infection paths leading from an initial viral state $v=(v_1, \ldots, v_N)$ to the final state $v^{(l)}=(m_1, \ldots, m_N)$.

Theorem: The following formula holds for the number of infection paths:

$$I_N^{(v)} = \prod_{i=1}^{N-1} \binom{\sum_{k=i}^{N} (m_k - v_k)}{m_i - v_i}$$

Proof: Whenever a program is infected one writes down the number of the account in which it resides. Thus the infection process is represented as a finite sequence of integers with elements between 1 and N. The length of the sequence is

$$\sum_{i=1}^{N} (m_i - v_i)$$

There are exactly $m_i - v_i$ entries with value i in this sequence for $1 \leq i \leq N$. This gives for $N=2$

$$I_2^{(v)} = \binom{m_1 + m_2 - v_1 - v_2}{m_1 - v_1}$$

The rest follows by induction on N.

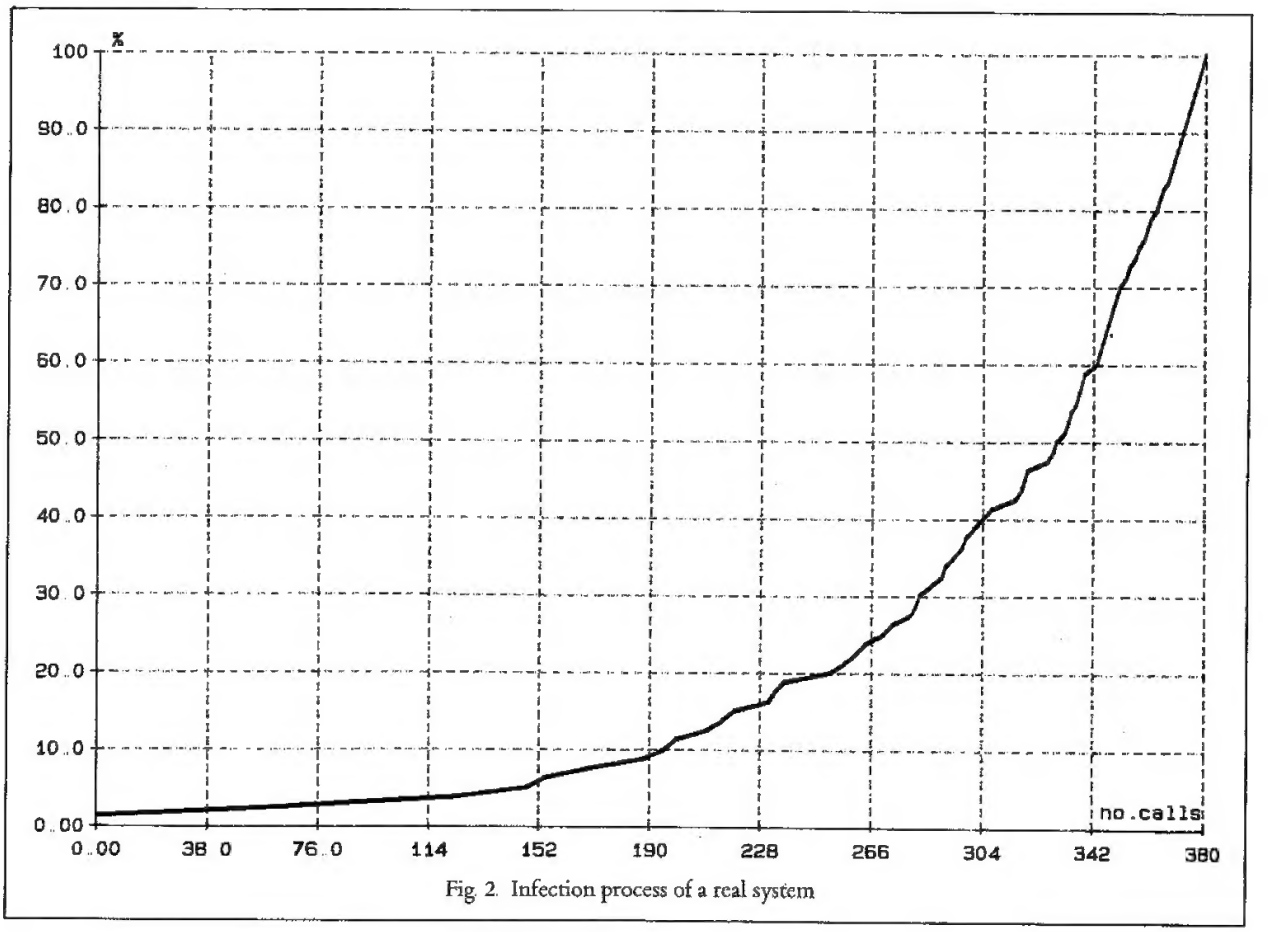


Fig. 2. Infection process of a real system

This shows that it is not feasible to do calculations for more than one account and a sufficiently large number of programs.

One account with m programs: For convenience the assumption is made that all programs are called with equal frequency $1/m$. The computation proceeds as follows. Initially only one program is infected. After each call of a program the expectation for the number of infected programs is calculated. As soon as it exceeds the number of initially infected programs by more than one, the process is started anew but with one more program, which is initially infected.

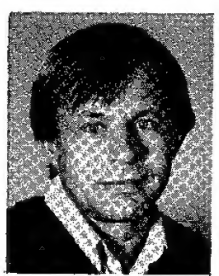
On the x axis the number of runs of a program is plotted and on the y axis the percentage of infected programs. Supposing that there are 80 programs of which only one is infected in the beginning the model shows that all of them are infected after 378 calls (Fig. 1).

This is contrasted with an infection process on a real system, where the number of the program to be called was determined by a random number generator. Figure 2 shows a typical infection process on a real system. The number of program calls, after which the whole account was infected, lay in the range from 230 and 700. The nature of the curve shows an exponential growth of the infection process for the real system as well as for the probability model. Assuming an average number of ten runs per hour this shows that after about 40 hours all programs are infected. The computations show that m programs will be infected after approximately $5m$ runs.

Takeover of users: The model can be used to say something when all users are infected instead of the takeover of all programs. This problem can be treated with similar arguments as above. Assuming that each account has only one program or that the virus infects all programs of the account from which it was called, this case is reduced to the situation discussed earlier.

References

[1] F. Cohen, *Computer Viruses, Theory and Experiments, Preprint*, University of Southern California, 1984.
[2] F. Cohen, Computer viruses, *PhD Thesis*, University of Southern California, 1985
[3] S. Lang, *Linear Algebra*, Addison-Wesley, Reading, MA, 1970



Dr. W. Gleissner received a Masters degree in numerical analysis from Oxford University in 1972 and a Diploma in mathematics from the Technical University of Munich in 1973. His main fields of interest were interval arithmetic and approximation theory but later changed to pure mathematics and representation theory of infinite groups. He received a doctoral degree in 1978 from the Technical University of Munich. He then joined the Munich Reinsurance Co. as a project leader for the computer-based administration of the life reinsurance business for client insurance companies from all over the world. Later he was in charge of introducing workstations and personal computers in the same company.

In 1985 he became responsible for guideline development and programming methodology in an institution producing software for the administration of the Federal Republic of Germany. His main fields of interest now are computer viruses and security-related topics in Unix. He has published more than a dozen papers about mathematical models in economics. (Industrieanlagen-Betriebsgesellschaft mbH, Einsteinstrasse 20, D-8012 Ottobrunn, F.R.G.)